# Artificial Intelligentt Method for Tuning the Output Scaling Factor of a Fuzzy Controller

**Fakhrulddin H. Ali *** Fhali310@yahoo.com
**Mohammed M. Hussein*** mohmah86@gmail.com
**Sinan M.B. Ismael**** eng_sinan85@yahoo.com

* Computer Engineering Department, College of Engineering, University of Mosul, IRAQ
Computer and Information Eng. Dept., College of Electronics Eng., University of Mosul, Iraq

## Abstract

**Scaling factor tuning is one of the most used method to enhance the performance of a fuzzy controller. This paper presents two intelligent tuning strategies to tune this factor. In the first strategy, a supervisor fuzzy controller SFC was designed to continuously adjust, on line, the scaling factor of the basic fuzzy controller BFC based on the error and change of error signals. In the second strategy, a neural network NN is used to do this task. Performance of the tuning strategies are compared with corresponding conventional fuzzy controller in terms of several performance measures such as steady state error, settling time, rising time, and peak overshoot. Simulation results show that SFC performance is better. The system implementation and tests are carried out using LabVIEW (V 8.2).**

**Keywords: Artificial Intelligent, Fuzzy Controller, Neural Network, LabVIEW.**

# استخدام طريقة الذكاء الاصطناعي لتنغيم معامل تقييس إخراج المسيطر المُضبّب

**فخر الدين حامد علي**
قسم هندسة الحاسبات
كلية الهندسة

**محمد محمود حسين**
قسم هندسة الحاسوب
كلية الهندسة
جامعة الموصل – العراق

**سنان محمد بشير إسماعيل**
قسم هندسة الحاسوب والمعلوماتية
كلية هندسة الالكترونيات

## الخلاصة

**إن عملية تنغيم معامل تقييس الإخراج تعتبر واحدة من أكثر الطرائق استخداماً لتحسين أداء المسيطر المُضبّب. حيث يقدم هذا البحث آلية استخدام إستراتيجيتين من استراتيجيات الذكاء الاصطناعي في تنغيم هذا المعامل. تضمنت الإستراتيجية الأولى تصميم مسيطر مُضبّب مشرف لتعديل معامل تقييس الإخراج باستمرار أثناء العمل للمسيطر المُضبّب الرئيس بالاعتماد على إشارتي الخطأ والتغير في الخطأ. أما الإستراتيجية الثانية فتضمنت استخدام خوارزمية الشبكة العصبية للقيام بهذا الغرض. قُورن أداء استراتيجيات التوليف مع أداء المسيطر المُضبّب التقليدي أي عندما تكون قيمة معامل التقييس ثابتة وبالاعتماد على العديد من مقاييس الأداء مثل خطأ حالة الثبوت, زمن الارتفاع, زمن الاستقرار, النسبة المئوية لتجاوز الحد. إذ أثبتت نتائج المحاكاة أن استخدام المسيطر المُضبّب المشرف في عملية التنغيم يكون أفضل. إن تنفيذ تصاميم النظام واختباره تم في بيئة LabVIEW ذي الإصدار 8.2.**

Received: 15– 3 - 2010 Accepted: 24 – 11 - 2010

## 1. Introduction:

Fuzzy logic controllers FLCs have rapidly gained popularity in many engineering disciplines after the concept of fuzzy logic was introduced by Zadeh and after Mamdani and his coworkers presented a fuzzy control scheme [1]. The main advantage of fuzzy logic as compared to the conventional control approach reside in the fact that no mathematical modeling is required for the design of the controller. The controller rules are based especially on the knowledge of the system behavior and on the experience of the control engineer [2].

Since the inception of FLCs, investigating effective tuning method of FLCs has been an active research area [1]. For the successful design of FLC's proper selection of input and output scaling factors (SF's) and/or tuning of the other controller parameters are crucial jobs, which in many cases are done through trial and error or based on some training data [3]. Basically, the control performance of the FLCs can be enhanced by the following ways: modifying membership functions [4], inference mechanism improving [5], rule tuning [6] and scaling factors adjusting [7] [8]. Out of the various tunable parameters, SF's have the highest priority due to their global effect on the control performance. However, relative importance of the input and output SF's to the performance of a fuzzy logic control system is yet to be fully established.

Focused on the tough job of finding proper scaling factors especially output scaling factor in the control process, this paper presents an intelligent tuning strategies for acquiring the right output scaling factor.

## 2. Mathematical model of D.C. servo motor:

Consider the speed control dc servomotor shown in fig. (1) [9], where the field current is held constant. In this system the parameters of the system are given in the table 1 [10]. The open loop transfer function of the system is obtained by the following equation:

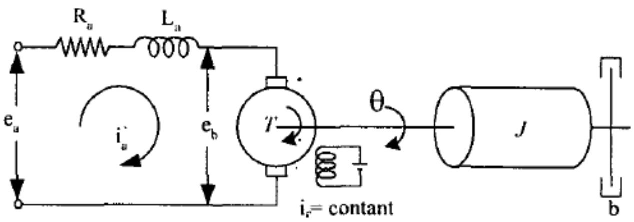



**Figure 1: Schematic diagram of armature-controlled dc servo motor**

$$G(s) = \frac{\theta_m(s)}{E_a(s)} = \frac{AK_i n}{L_a J_{me} s^3 + (J_{me} R_a + L_a B_{me}) s^2 + (R_a B_{me} + K_b K_i) s} \quad \ldots(1)$$

*Where*

$$J_{me} = J_m + J_L n^2 \quad \ldots(2)$$

$$B_{me} = B_m + B_L n^2 \quad \ldots(3)$$

Suppose that the inductance La=0.1 Henry, the open loop transfer function of the third order model is:

$$G(s) = \frac{250\ A}{s(s^2 + 50.5\ s + 1725)} \quad \ldots(4)$$

The inductance La in the armature circuit is usually small and may be neglected[9]. If La is neglected, then the open loop transfer function of the second order model becomes:[10]

$$G(s) = \frac{5A}{s(s+34.4)} \qquad \ldots(5)$$

**Table 1: Positional control system parameter**

| Parameters | Description | Value |
|---|---|---|
| $K_i$ | torque constant | 0.5 Ib-ft/amp |
| $A$ | DC amplifier gain | 10 |
| $R_a$ | armature resistance | 5Ω |
| $J_m$ | motor shaft moment of inertia | $10^{-3}$ Ib-ft-sec $^2$ |
| $B_m$ | motor shaft friction | Negligible |
| $B_L$ | load shaft friction | 0.1 Ib-ft-sec $^2$ |
| $J_L$ | load moment of inertia | 0.1 Ib-ft-sec $^2$ |
| $n$ | gear ratio | 1/10 |
| $K_b$ | back *e.m.f.* constant | 1.356Ki |

## 3. Basic fuzzy controller design:

The design of a fuzzy logic controller needs the selections of such control elements and parameters as scaling factors for input/output signals, a set of rule base, fuzzification and defuzzification methods and operations for the fuzzy reasoning, which include an implication, a compositional and aggregation operations of antecedents and consequents [11]. The LabVIEW software is used to design the controller using fuzzy logic toolkit which enables integrating a fuzzy controller into virtual instrument environment. This controller has been designed with two inputs position error (e(k)) and change of position error ( $\Delta$ e(k)) and a single output (u(k)) representing the control signal to the motor as shown in fig. (2).

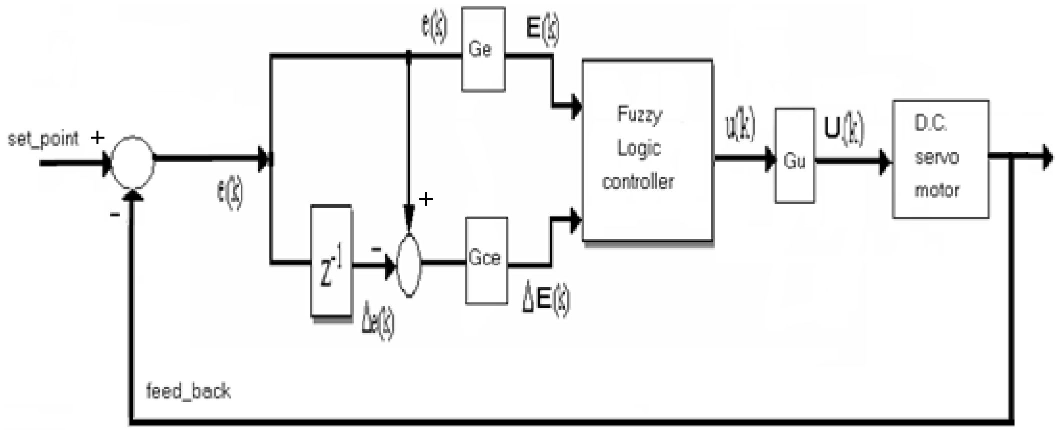



**Figure 2: Fuzzy controller scheme**

The seven triangular input and output membership functions are adopted for the controller are shown in the fig. (3). For the system under study the universe of discourse for both e(k) and $\Delta$ e(k) may be normalized to be within the range [-1,1], and the linguistic labels are {NB, NM, NS, ZE, PS, PM, PB}.

The controller action is based on Mamdani fuzzy type, center of gravity defuzzification method, min-max inference engine. The rules which states the relationship between the input

domain fuzzy sets and output domain fuzzy sets can be derived from a step response curve of a closed-loop system and are represented in a tabular form as shown in table 2. Fig. (4) depicts a control surface of the basic fuzzy controller [12].

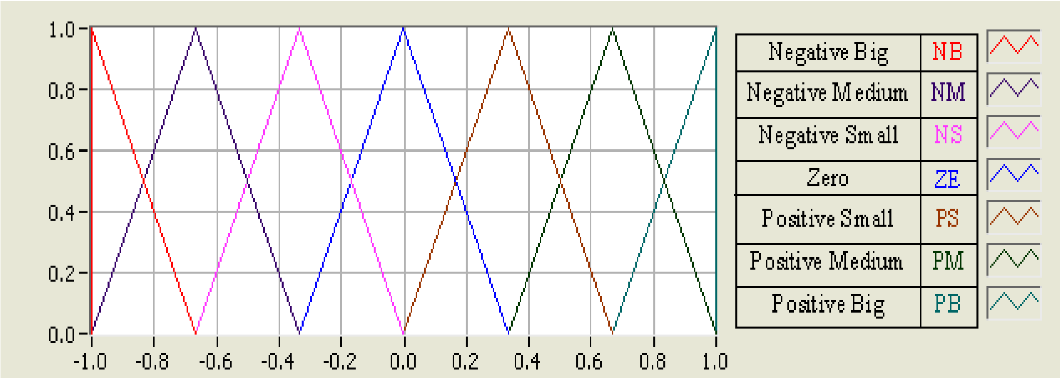



**Figure 3: Membership function for E, △ E, and u**

**Table 2: Rule base for basic fuzzy controller**

| **E** / **ΔE** | **NB** | **NM** | **NS** | **ZE** | **PS** | **PM** | **PB** |
|---|---|---|---|---|---|---|---|
| **NB** | NB | NB | NB | NM | NS | NS | ZE |
| **NM** | NB | NM | NM | NM | NS | ZE | PS |
| **NS** | NB | NM | NS | NS | ZE | PS | PM |
| **ZE** | NB | NM | NS | ZE | PS | PM | PB |
| **PS** | NM | NS | ZE | PS | PS | PM | PB |
| **PM** | NS | ZE | PS | PM | PM | PM | PB |
| **PB** | ZE | PS | PS | PM | PB | PB | PB |

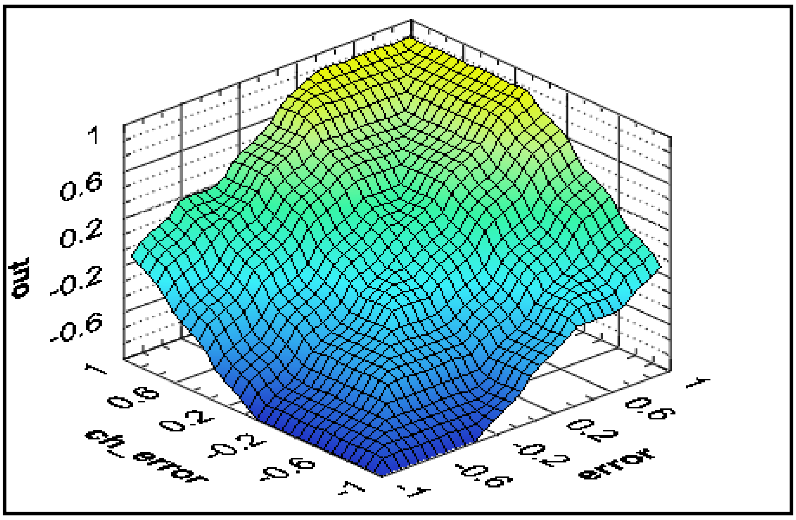



**Figure 4: Control surface for basic fuzzy controller**

The adjustment of scaling factor Gu affects the output control of the basic fuzzy controller directly. If Gu increases, the universe of discourse of output variable will extend and the actual value of the output of the controller will increase too, and thus the rise time will decrease and the transient response will be fast. But a too large Gu may cause the overshoot and oscillation as shown in fig. (5) and table 3. On the other hand, decreasing Gu is benefit to the stability of the system but it will make the transient response slow.

**Table 3: Comparison result for effect of Gu on the response**

| Gu | Overshoot (%) | Settling time (S) | Rising time (S) |
|---|---|---|---|
| **5** | 0 | 0.54 | 0.22 |
| **10** | 20.42 | 0.48 | 0.05 |
| **15** | 59.39 | 1.32 | 0.04 |

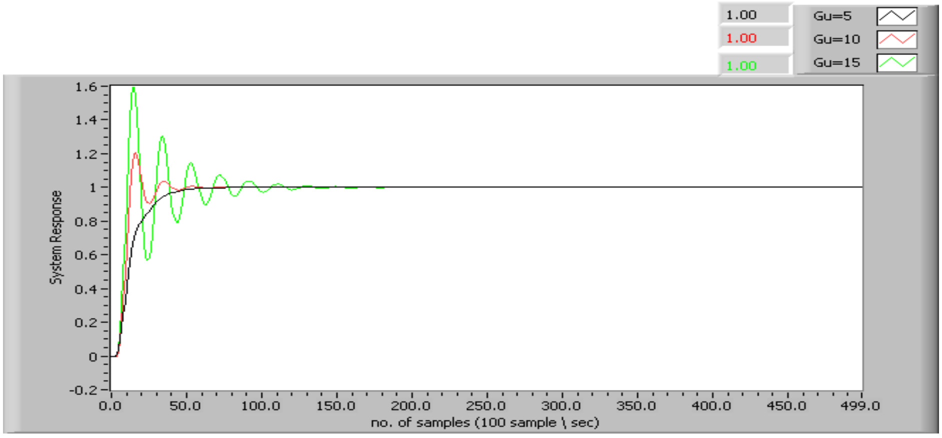



**Figure 5: Effect of Gu on the response**

## 4. Self tuning fuzzy logic controller:

Fuzzy logic controller (FLC) is called adaptive if any one of its tunable parameters (scaling factors, membership functions, and rules) changes when the controller is being used, otherwise it is a not adaptive or conventional FLC. An adaptive FLC that is tuned by modifying either its membership functions or scaling factors or both of them is called a self-tuning FLC [3]. On the other hand, when a FLC is tuned by automatically changing its rules then it is called a self-organizing FLC [3][13]. Since the presented FLC is tuned by modifying the output gain depending on the present value of error and change in error it described as a self-tuning FLC. The block diagram of the proposed self tuning FLC is shown in fig. (6).

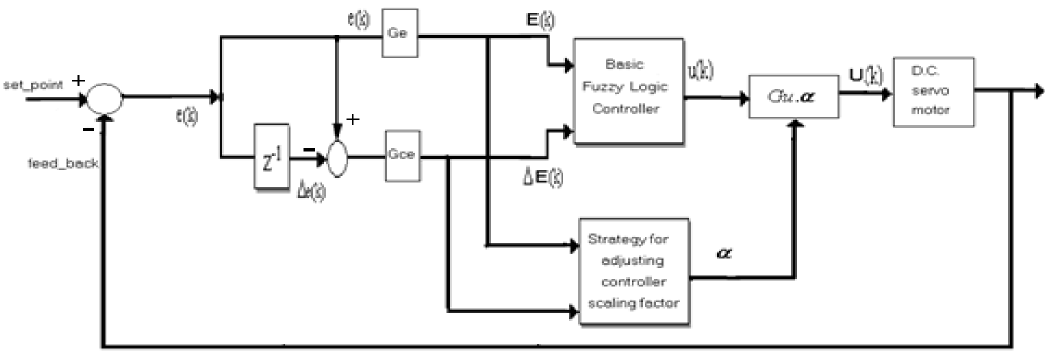



**Figure 6: Block diagram of the self-tuning FLC**

The value of Gu is constant for a particular type of conventional FLC. But the gain of the self-tuning FLC does not remain fixed while the controller is in operation, rather it is modified in each sampling time by the gain updating factor(α), depending on the trend of the controlled process output. The reason behind this on-line gain variation is to make the controller respond according to the desired performance specifications

### 4.1 Supervisor fuzzy controller strategy:

In this controller type, the output scaling factor of the controller is adjusted, on line, by a SFC. The SFC is a two dimension FLC with two inputs (e, Δe) and one output (α). The membership functions for controller inputs are defined on the common normalized domain [-1, +1] as shown in fig. (3) whereas the membership functions for the gain factor (α) is defined on the normalized domain [0, +1] as shown in fig. (7). The rule base in table 4 is used for the computation of $\alpha$ . Fig. (8) depicts a control surface of the supervisor fuzzy controller. The relationships between the output gain and the output variables of the self tuning fuzzy controller becomes as in equation (6).

$$U(k) = Gu \bullet \alpha \times u(k) \quad \ldots(6)$$

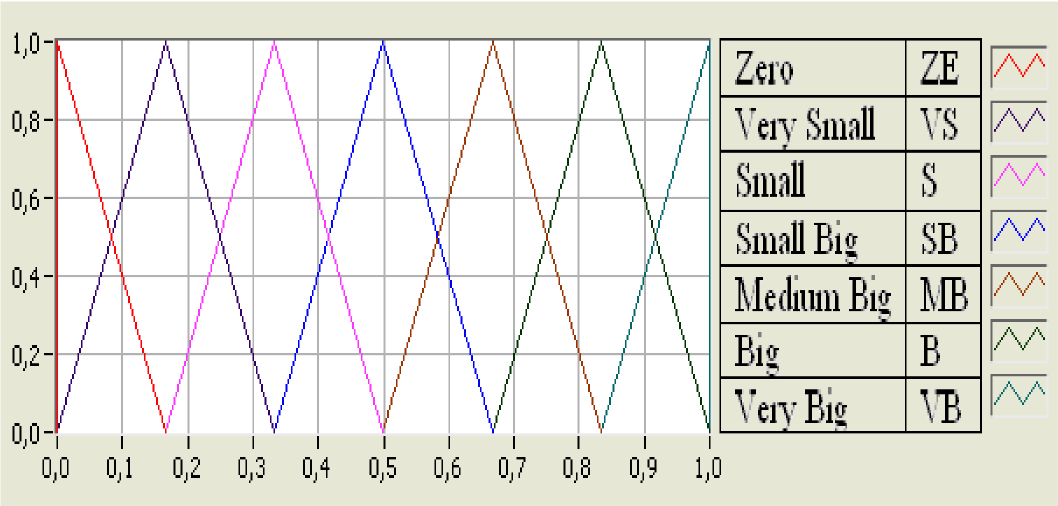



**Figure 7: Member ship function for $\alpha$**

**Table 4: Rule Base for $\alpha$**

| E / ΔE | NB | NM | NS | ZE | PS | PM | PB |
|---|---|---|---|---|---|---|---|
| **NB** | VB | VB | VB | B | SB | S | ZE |
| **NM** | VB | VB | B | B | MB | S | VS |
| **NS** | VB | MB | B | VB | VS | S | VS |
| **ZE** | S | SB | MB | ZE | MB | SB | S |
| **PS** | VS | S | VS | VB | B | MB | VB |
| **PM** | VS | S | MB | B | B | VB | VB |
| **PB** | ZE | S | SB | B | VB | VB | VB |

### 4.2 Neural network strategy:

Neurons are the essential blocks of the neural network. They are processing elements which accept the input signals, process them using a certain function and then develop an output signal. There are weight and bias associated with each neuron [14].

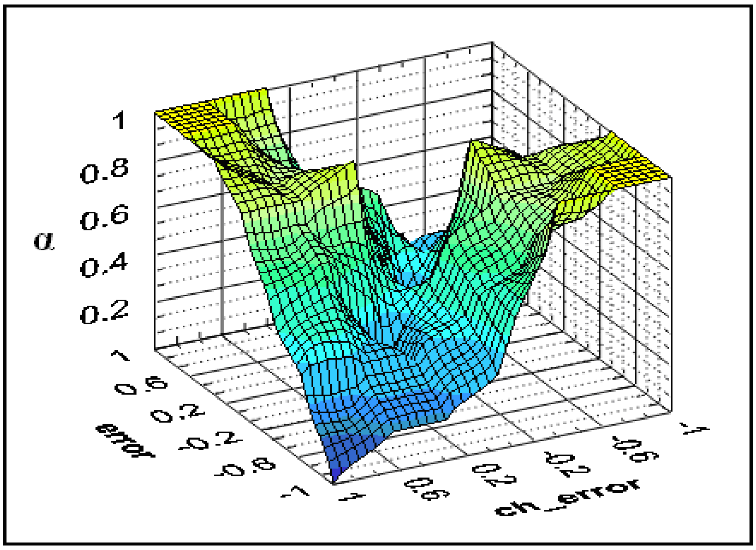



**Figure 8: Control surface for supervisor fuzzy controller**

An Artificial Neural Network (ANN) is a collection of neurons in a special configuration. It is a parallel processor that can represent function which has large number of inputs and in which the relation between the inputs and the outputs cannot be easily written as a mathematical equation. Basically in ANN, knowledge is stored in memory as experience and is available for use at future time. The weights and biases associated with the output of each neuron represent the memory which stores the knowledge. Each neuron may function locally by itself; collection of neurons work together to estimate the required function [14].

Feed forward back propagation ANN consists of two processing parts within its neurons: forward and backward. As any input is fed to the ANN during its training process, the ANN will try to learn and compares its predicted output values with the desired value. The errors between the predicted and the actual value are then back propagated through the network, and a gradient descent algorithm is used to adjust the weights in the hidden and output layer nodes. The result is a network that produces the required mapping between the input values and the output values in the neurons [15].

This technique is used to implement the control structure. Such capability allows the network to learn, and so adjusts the output scaling factor of the controller, online. The system characteristics are defined through multilayer Back propagation Neural Network (BPNN). The output of the trained network is used to adjust the output scaling factor in real time.

A two layers feed-forward network with both sigmoid hidden neurons and linear neurons are constructed in LabVIEW . Training by back-propagation involves three stages: feed-forward, back propagation on errors and weight updating. The output of one layer becomes the input to the following layer.

With different values of e and Δe, the correct value of α is obtained by using αNN's, which is a two layers BPNN's with a structure of 2-9-1 as shown in fig. (9).

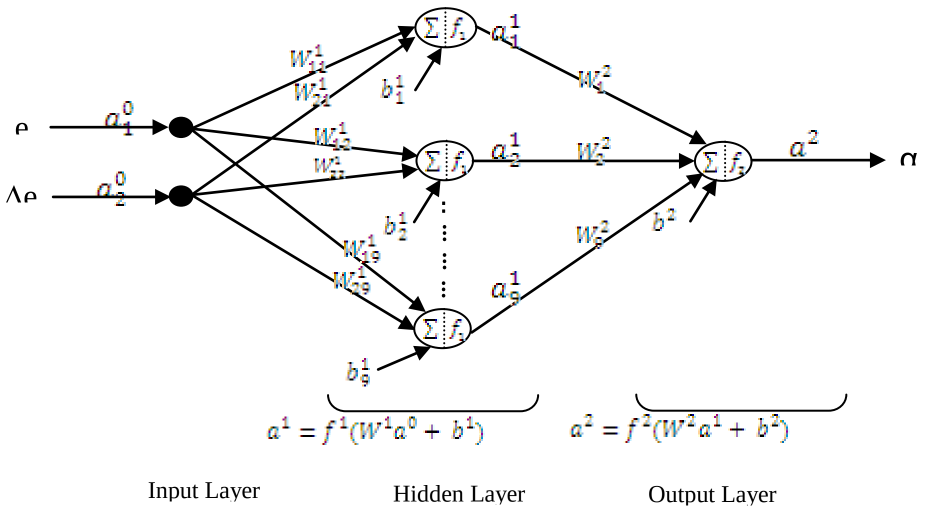



**Figure 9: Block diagram of αNN's**

The forward computation formulas of αNN's are as follows [16]:
The first step is to propagate the input forward through the network.

$$a^{m+1} = f^{m+1}\left(W^{m+1}a^m + b^{m+1}\right)\ldots \quad \text{for m} = 0,1 \ldots\ldots \mathrm{M}-1 \qquad \ldots(7)$$

Where M is the number of layers in the network, in the current network M=2. The neuron in the first layer receive the external inputs:

$$a_1^0 = e\ , a_2^0 = \Delta e$$

This provides the starting point for eq. 7. The output of the neuron on last layer is considered the network output:

$$\alpha = a^M, in\ the\ current\ network\ \alpha = a^2$$

The next step is to propagate the sensitive, s, backwards through the network:

$$s^M \rightarrow s^{M-1} \rightarrow \ldots \rightarrow s^2 \rightarrow s^1$$

This is obtained at final layer

$$s_i^M = \frac{\partial F}{\partial n_i^M} = \frac{\partial (t-a)^T (t-a)}{\partial n_i^M} = \frac{\partial \sum_{j=1}^{S^M}(t_j - a_j)^2}{\partial n_i^M} = -2\left(t_j - a_i\right)\frac{\partial a_i}{\partial n_i^M} \qquad \ldots(8)$$

Now, since

$$\frac{\partial a_i}{\partial n_i^M} = \frac{\partial a_i^M}{\partial n_i^M} = \frac{\partial f^M(n_i^M)}{\partial n_i^M} = f^M(n_i^M) \qquad \ldots(9)$$

$$\Rightarrow s_i^M = -2(t_i - a_i) f^M(n_i^M) \qquad \ldots(10)$$

This can be expressed in matrix form as:

$$\Rightarrow s^M = -2F^M(n^M)(t-a) \qquad \ldots(11)$$

$$\Rightarrow s^m = -2F'^m(n^m)(W^{m+1})^T S^{m+1}, for\ m = M-1, \ldots .2,1 \qquad \ldots(12)$$

Finally, the weights and biases are updated using approximation steepest descent rule:

$$W^M(K+1) = W^m(k) - \alpha_r\, s^m (a^{m-1})^T \qquad \ldots(13)$$

where $\alpha_r$ is the learning rate.

$$b^m(K+1) = b^m(k) - \alpha_r s^m \qquad \ldots(14)$$

## 5. Simulation results:

The simulation results presented in this section are for third and second order representation of positional control system models. The performance of self tuning strategies are compared with conventional fuzzy controller. In order to make a comparison between self tuning and fixed output gain fuzzy controllers, several performance parameters such as overshoot, settling time, rising time, and steady state error are measured. The step response is shown in fig. (10), for the second order model, and in fig. (11), for the third order model. The parameters values are tabulated in table (5) and (6) for second and third order model respectively. Fig. (12) and (13) demonstrate the response to a square wave input for second and third order models respectively.

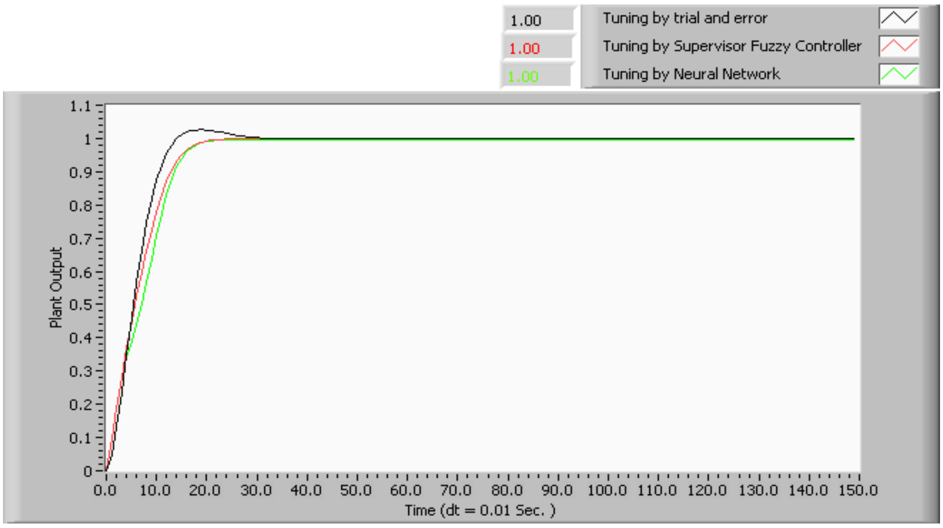



**Figure 10: Second order model step response**

**Table 5: Comparison of response characteristics for figure (10)**

| Tuning strategy | Overshoot (%) | Settling time (S) | Rising time (S) |
|---|---|---|---|
| **Trial and error** | 2.6 | 0.28 | 0.09 |
| **Supervisor Fuzzy Controller** | 0 | 0.21 | 0.11 |
| **Neural Network** | 0 | 0.23 | 0.12 |

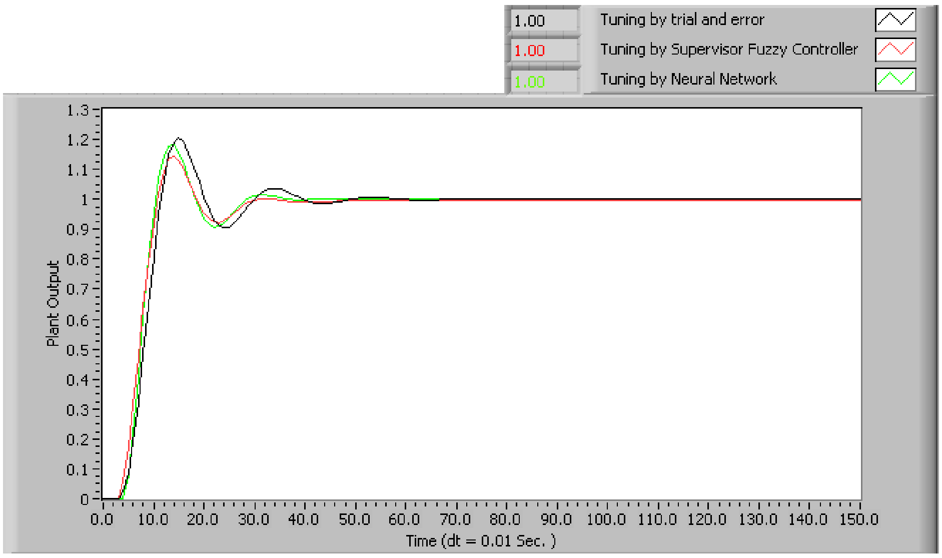



**Figure 11: Third order model step response**

**Table 6: Comparison of response characteristics for figure 11**

| Tuning strategy | Overshoot (%) | Settling time (S) | Rising time (S) |
|---|---|---|---|
| **Trial and error** | 20.42 | 0.48 | 0.05 |
| **Supervisor Fuzzy Controller** | 14.26 | 0.30 | 0.05 |
| **Neural Network** | 17.94 | 0.34 | 0.04 |

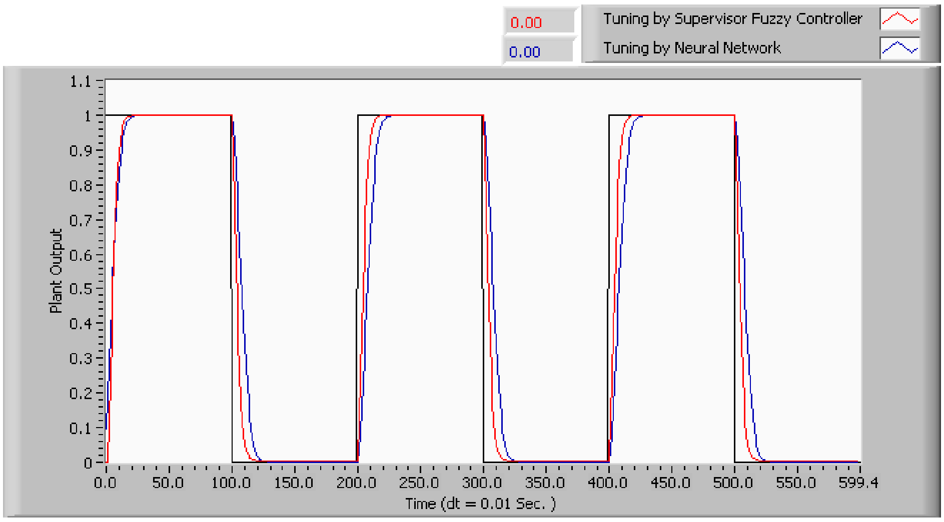



**Figure 12: System response to square wave input for the second order model**

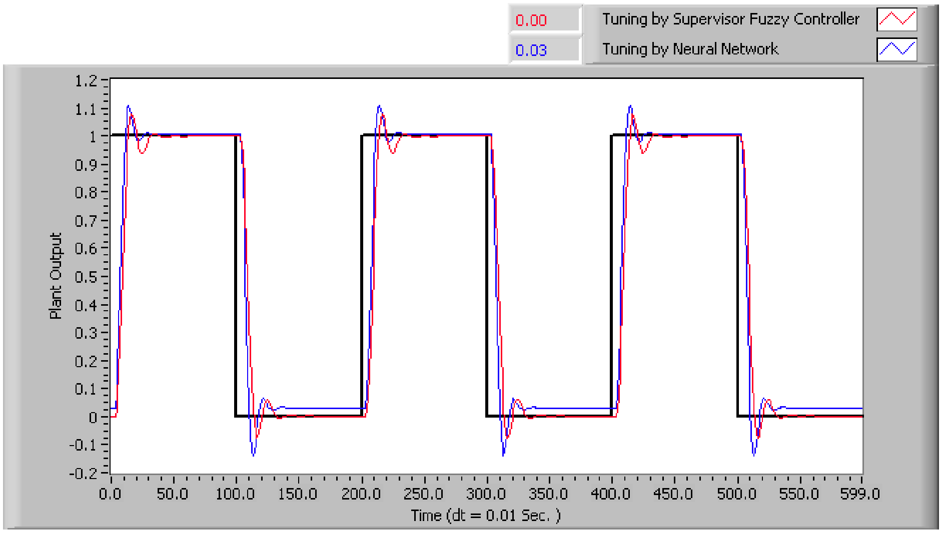



**Figure 13: System response to square wave input for the third order model**

Fig. (14) and (15) shows the system response when (0.5 N.m) load is applied and released for second and third order models respectively. Table (7) and (8) summarize the drop, and overshoot when load is applied and released respectively.

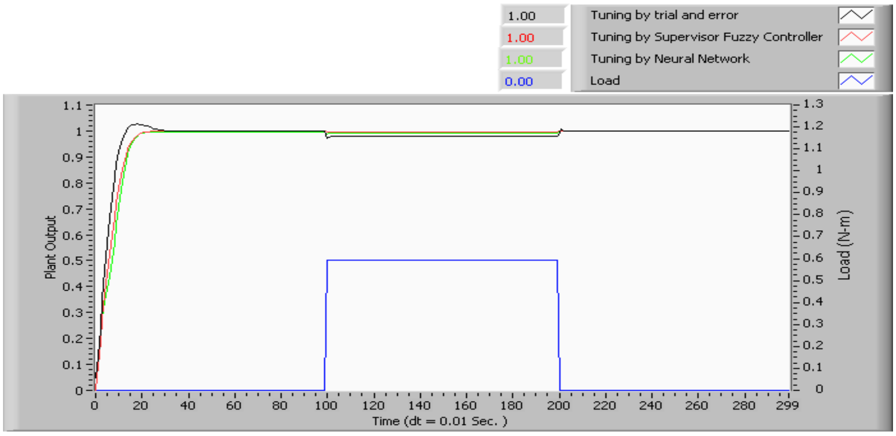



**Figure 14: Second order model step response when load is applied and released**

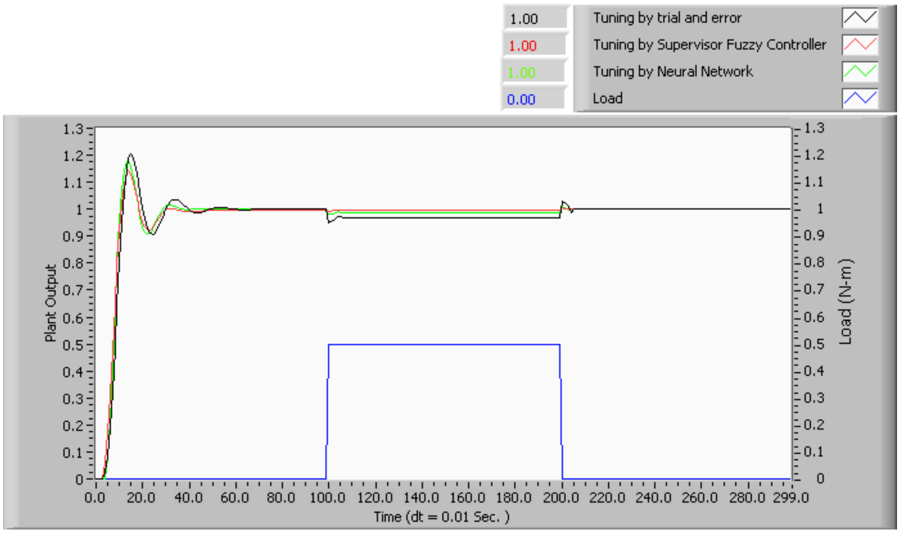



**Figure 15: Third order model step response when load is applied and released**

**Table 7: Response drop when 0.5 N-m load applied**

| Tuning strategy | Drop (%) 2nd Order System | Drop (%) 3rd Order System |
|---|---|---|
| **Trial and error** | 3.21 | 5.13 |
| **Supervisor Fuzzy Controller** | 0.57 | 1.02 |
| **Neural Network** | 1.67 | 2.22 |

**Table 8: Response overshoot when 0.5 N-m load released**

| Tuning strategy | Overshoot (%) 2nd Order System | Overshoot (%) 3rd Order System |
|---|---|---|
| **Trial and error** | 0.8 | 3.51 |
| **Supervisor Fuzzy Controller** | 0 | 0.52 |
| **Neural Network** | 0 | 1.13 |

## 6. Conclusions:

Due to the fact that tuning the gain of the FLC is difficult to set with iterative manual process, a fuzzy controller with an intelligent tuning strategies of the output scaling factor is proposed in this paper. This approach decreases the effective tuning time in addition to the fact that it may consider any future variations in system parameters. However, the goal is to compare two methods and concludes which is better to implement adaptive FLC which is the main contribution of this paper. It can be seen from the simulation test results that the self tuning fuzzy controller with SFC strategy has better performance than NN strategy and conventional fuzzy controllers for both second and third order models.

## References:


**[1]** H. Zhuang, S. Wongsoontorn, **"Design and Tuning of Fuzzy Control Surfaces with Bezier Functions",** IEEE International Conference on Systems**,** Man and Cybernetics, Volume: 3, On page(s): 2194- 2199, 2005.

**[2]** N. Khongkoom, A. Kanchanathep, S. Nopnakeepong, S. Tanuthong, S. Tunyasrirut, R. Kagawa, **"Control of the Position DC Servo Motor by Fuzzy Logic",** Proceedings TENCON, vol.3, pp:354 – 357, 2000.

**[3]** Rajani K. Mudi and Nikhil R. **"A Robust Self-tuning Scheme for PI- and PD-Type Fuzzy Controllers",** IEEE transactions on fuzzy systems, VOL. 7, page(s): 2-16, Feb. 1999.

**[4]** I. Rojas, H. Pomares, J. Gonzalez, L.J. Herrera, A. Guillen, F. Rojas, O. Valenzuela**, "Adaptive fuzzy controller: Application to the control of the temperature of a dynamic room in real time",** Fuzzy Sets and Systems, Volume 157, Issue 16, Pages 2241-2258, 16 August 2006.

**[5]** F. Herrera, F.A. Marquez, A. Peregrin, **"Genetic adaptation of rule connectives and conjunction operators in fuzzy rule based systems: an experimental comparative study",** Proc. Third Internat. Conference of the European Society for Fuzzy Logic and Technology, Zittau, Germany, pp. 100--104, Sep. 2003.

**[6]** Hsuan-Ming Feng and Ching-Chang Wong, **"A On-line Rule Tuning Grey Prediction Fuzzy Control System Design",** Proceedings of the 2002 International Joint Conference on Neural Networks IJCNN, Honolulu, HI, page(s): 1316 – 1321, 07 August 2002.

**[7]** Zhiqiang G. Thomas A. and James G. **"A Stable Self-tuning Fuzzy Logic Control System for Industrial Temperature Regulation",** IEEE transaction on industry applications**,** VOL. 38 , NO.2 , March 2002**.**

**[8]** Mokrani L. and Kouzi K. **"Influence of Fuzzy Adapted Scaling Factors on the Performance of A Fuzzy Logic Controller Based on an Indirect Vector Control for Induction Motor Drive" ,** Journal of Electrical Engineering, VOL. 55, NO. 7-8, 188-194. ISSN 1335-3632 , FEI STU 2004.

**[9]** Katsuhiko Ogata, "Modern Control Engineering", Prentice Hall International, Inc., ISBN 0-13-590258-4, 1970.

**[10]** Benjamin C. Kuo, **"Automatic Control Systems",** Prentice Hall, Inc., ISBN 0-13-054973-8, third edition, 1975.

**[11]** L. Reznik, **"Fuzzy Controller Design: Recommendations to the User"**, Second Internation Conference on Knowledge-Based Intelligent Electronic Systems, Adelaide, Australia, Volume 3, Page(s):609 - 616, Apr. 1998.

**[12]** X. Tian, X. Wang, and Y. Cheng, **"A Self-tuning Fuzzy Controller for Networked Control System"**, International Journal of Computer Science and Network Security IJCSNS, VOL.7, Jan. 2007.

**[13]** S. Mohan and S. Bhanot, **"Comparitive Study of Some Adaptive Fuzzy Algorithms for Manipulator Control",** International Journal of Computational Intelligence, Volume **3** Number 4. pp. 303–311, 2007.

**[14]** Pogula Sridhar, Sriram **"Developing Neural Network Applications Using LabVIEW",** M. SC. Thesis, University of Missouri-Columbia, July 2005.

**[15]** Soo-See Chai, Bert Veenendaal, Geoff West, Jeffrey Philip Walker, **"Backpropagation Neural Network For Soil Moisture Retrieval Using NAFE’05 Data : A Comparison Of Different Training Algorithms",** The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B4. Beijing, china, 2008.

**[16]** Martin T. Hagan, Howard B. Demuth, Mark Beale, **"Neural Network Design",** ISBN: 7-111-10841-8, 1996 by PWS Publishing Company.

**The work was carried out at the college of Engineering. University of Mosul**